

CLAIMS

- 1 1. An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 a database table residing in the memory;
5 a query residing in the memory that accesses the database table;
6 a query optimizer residing in the memory and executed by the at least one
7 processor, wherein the query optimizer analyzes the query, and if no optimal index for the
8 query exists, the query optimizer determines if a sub-optimal index exists, a sub-optimal
9 index including at least one key referenced in the query and additionally including at least
10 one additional key that prevents traversal of the sub-optimal index to determine the
11 number of rows that the query will return, wherein if a sub-optimal index exists, the query
12 optimizer uses statistical information for each additional key to bridge any gaps in the
13 sub-optimal index when probing the sub-optimal index to optimize the query.
- 1 2. The apparatus of claim 1 wherein the statistical information comprises a frequent
2 values list that corresponds to an additional key.
- 1 3. The apparatus of claim 2 wherein the query optimizer bridges any gaps in the sub-
2 optimal index by rewriting the query by adding a reference to each additional key with
3 values that are taken from the frequent values list corresponding to the additional key.
- 1 4. The apparatus of claim 1 wherein the query optimizer probes the sub-optimal
2 index to determine an estimated number of rows in the database that satisfy the query.

- 1 5. An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 a database table residing in the memory;
5 a query residing in the memory that accesses the database table; and
6 a query optimizer residing in the memory and executed by the at least one
7 processor, wherein the query optimizer analyzes the query, and if no optimal index for the
8 query exists, the query optimizer determines if a sub-optimal index exists, a sub-optimal
9 index including at least one key referenced in the query and additionally including at least
10 one additional key that prevents traversal of the sub-optimal index to determine the
11 number of rows that the query will return, wherein if a sub-optimal index exists, the query
12 optimizer, for each additional key in the sub-optimal index, reads statistical information
13 regarding the additional key, the query optimizer rewriting the query using the statistical
14 information in a manner that allows probing the sub-optimal index according to the
15 rewritten query, the query optimizer thereby determining from the sub-optimal index an
16 estimated number of rows in the database table that satisfy the query.
- 1 6. The apparatus of claim 5 wherein the statistical information comprises a frequent
2 values list that corresponds to an additional key.

- 1 7. An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 a database table residing in the memory;
5 a query residing in the memory that accesses the database table;
6 an index residing in the memory that includes at least one key referenced in the
7 query and additionally includes at least one additional key that prevents traversal of the
8 index to determine the number of rows that the query will return; and
9 a query optimizer residing in the memory and executed by the at least one
10 processor, wherein the query optimizer, for each additional key in the index, reads
11 statistical information regarding the additional key, the query optimizer rewriting the
12 query using the statistical information in a manner that allows probing the index
13 according to the rewritten query, the query optimizer thereby determining from the index
14 an estimated number of rows in the database table that satisfy the query.
- 1 8. The apparatus of claim 7 wherein the statistical information comprises a frequent
2 values list that corresponds to an additional key.

1 9. A method for optimizing a database query for a database table, the method
2 comprising the steps of:
3 evaluating the query;
4 determining whether an optimal index for the query exists;
5 if no optimal index exists, determining whether a sub-optimal index for the query
6 exists, a sub-optimal index including at least one key referenced in the query and
7 additionally including at least one additional key that prevents traversal of the sub-
8 optimal index to determine the number of rows that the query will return; and
9 if a sub-optimal index exists, using statistical information for each additional key
10 to bridge any gaps in the sub-optimal index when probing the sub-optimal index to
11 determine an estimated number of rows in the database that satisfy the query.

1 10. The method of claim 9 wherein the statistical information comprises a frequent
2 values list that corresponds to an additional key.

1 11. The method of claim 10 further comprising the step of bridging any gaps in the
2 sub-optimal index by rewriting the query by adding a reference to each additional key
3 with values that are taken from the frequent values list corresponding to the additional
4 key.

1 12. The method of claim 9 further comprising the step of:
2 if an optimal index exists, probing the optimal index to determine an estimated
3 number of rows in the database table that satisfy the query.

1 13. The method of claim 9 further comprising the step of:
2 if no optimal index exists and no sub-optimal index exists, performing a table
3 scan to determine a number of rows in the database table that satisfy the query.

- 1 14. A method for optimizing a database query for a database table, the method
2 comprising the steps of:
- 3 (1) analyzing the query;
- 4 (2) if no optimal index for the query exists, determining if a sub-optimal index
5 exists, a sub-optimal index including at least one key referenced in the query and
6 additionally including at least one additional key that prevents traversal of the sub-
7 optimal index to determine the number of rows that the query will return;
- 8 (3) if a sub-optimal index exists, performing the following steps for each
9 additional key in the sub-optimal index that prevents traversal of the sub-optimal index to
10 determine the number of rows that the query will return:
- 11 (A) reading statistical information regarding the additional key; and
12 (B) rewriting the query using the statistical information in a manner that
13 allows probing the sub-optimal index according to the rewritten query;
- 14 (4) probing the sub-optimal index using the rewritten query; and
15 (5) determining from the probe of the sub-optimal index an estimated number of
16 rows in the database table that satisfy the query.
- 1 15. The method of claim 9 wherein the statistical information comprises a frequent
2 values list that corresponds to an additional key.

- 1 16. A program product comprising:
2 (A) a query optimizer that analyzes a query for a database table, and if no optimal
3 index for the query exists, the query optimizer determines if a sub-optimal index exists, a
4 sub-optimal index including at least one key referenced in the query and additionally
5 including at least one additional key that prevents traversal of the sub-optimal index to
6 determine the number of rows that the query will return, wherein if a sub-optimal index
7 exists, the query optimizer uses statistical information for each additional key to bridge
8 any gaps in the sub-optimal index when probing the sub-optimal index to optimize the
9 query; and
10 (B) computer-readable signal bearing media bearing the query optimizer.
- 1 17. The program product of claim 16 wherein the computer-readable signal bearing
2 media comprises recordable media.
- 1 18. The program product of claim 16 wherein the computer-readable signal bearing
2 media comprises transmission media.
- 1 19. The program product of claim 16 wherein the statistical information comprises a
2 frequent values list that corresponds to an additional key.
- 1 20. The program product of claim 19 wherein the query optimizer bridges any gaps in
2 the sub-optimal index by rewriting the query by adding a reference to each additional key
3 with values that are taken from the frequent values list corresponding to the additional
4 key.

1 21. The program product of claim 16 wherein the query optimizer probes the sub-
2 optimal index to determine an estimated number of rows in the database that satisfy the
3 query.

- 1 22. A program product comprising:
- 2 (A) a query optimizer that analyzes a query for a database table, and if no optimal
- 3 index for the query exists, the query optimizer determines if a sub-optimal index exists, a
- 4 sub-optimal index including at least one key referenced in the query and additionally
- 5 including at least one additional key that prevents traversal of the sub-optimal index to
- 6 determine the number of rows that the query will return, wherein if a sub-optimal index
- 7 exists, the query optimizer, for each additional key in the sub-optimal index, reads
- 8 statistical information regarding the additional key, the query optimizer rewriting the
- 9 query using the statistical information in a manner that allows probing the sub-optimal
- 10 index according to the rewritten query, the query optimizer thereby determining from the
- 11 sub-optimal index an estimated number of rows in the database table that satisfy the
- 12 query; and
- 13 (B) computer-readable signal bearing media bearing the query optimizer.
- 1 23. The program product of claim 22 wherein the computer-readable signal bearing
- 2 media comprises recordable media.
- 1 24. The program product of claim 22 wherein the computer-readable signal bearing
- 2 media comprises transmission media.
- 1 25. The program product of claim 22 wherein the statistical information comprises a
- 2 frequent values list that corresponds to an additional key.

- 1 26. A program product comprising:
2 (A) a query optimizer that processes a query for a database table using a sub-
3 optimal index that includes at least one key referenced in the query and additionally
4 includes at least one additional key that prevents traversal of the sub-optimal index to
5 determine the number of rows that the query will return, wherein the query optimizer, for
6 each additional key in the index, reads statistical information regarding the additional key,
7 the query optimizer rewriting the query using the statistical information in a manner that
8 allows probing the index according to the rewritten query, the query optimizer thereby
9 determining from the index an estimated number of rows in the database table that satisfy
10 the query; and
11 (B) computer-readable signal bearing media bearing the query optimizer.
- 1 27. The program product of claim 26 wherein the computer-readable signal bearing
2 media comprises recordable media.
- 1 28. The program product of claim 26 wherein the computer-readable signal bearing
2 media comprises transmission media.
- 1 29. The program product of claim 26 wherein the statistical information comprises a
2 frequent values list that corresponds to an additional key.

* * * * *